

知の創成と検証支援システムの オープンソース化に伴う新機能の開発

2012/09/30

科学研究費補助金基盤(B) 22300036

学際的学問分野の BOK 策定を事例とした知の創成と検証支援システムの研究・開発

研究代表者：増永 義文

目次

[1. WikiBOK システム概要](#)

[BOK とは](#)

[アーキテクチャ](#)

[BOKEditor について](#)

[DescriptionEditor について](#)

[編集競合解消について](#)

[2. システム導入資料](#)

[事前確認](#)

[WikiBOK](#)

[PageEditToSNS](#)

[3. システム運用資料](#)

[編集操作について](#)

[ユーザ権限の昇格](#)

[空記事の表示について](#)

[記事の削除について](#)

[4. 新機能解説](#)

[ビジュアルライズモジュールの変更](#)

[Twitter への編集情報投稿](#)

【別紙】 WikiBOK 操作マニュアル

(計 11 ページ)

WikiBOK システム概要

WikiBOK はオープンソースの SemnaticMediaWiki(SMW)の拡張機能として開発されています。

主な機能としては、以下の通り。

多人数での共同編集により、知識体系を策定する「BOKEditor」

各記事の意味的リンク(SemanticLink)を可視化したまま、記事編集をする「DescriptionEditor」

各記事の編集を通知する「Twitter 連携」

BOK とは

BOK とは知識体系(Body Of Knowledge)の省略語です。

本システムでは、特定の単語(Node)とその詳細な記事(Document)を元に構成する階層関係をさします。

BOK は XML 構造と似た、ツリー構造をますが、各 Node の兄弟関係に順位がない点が異なります。

BOK は下記の特徴を持ちます。

上位 Node は下位 Node を包括する概念である

全く同じ名称の Node で、別の概念は存在しない

アーキテクチャ

本システムを利用するネットワーク概要を図 1-1 に示します。

本システムで利用しているプログラム/ライブラリの構成を図 1-2 に示します。

画面描画に d3.js ライブラリ、DOM 操作に jQuery ライブラリを利用しています。

これらのライブラリは、各クライアント上で動作するため、共同編集によるサーバ負荷を軽減します。

また、各操作の結果が即座に画面表示へ反映されるため、BOK における親子関係や意味的リンクの変更を視覚的にとらえ、データの構築を直感的に行うことができます。

本システムは、SemanticMediaWiki が動作している環境への導入を前提にしています。

しかしながら、サーバ環境によっては追加モジュールをインストールする必要があるかもしれません。

使用機能	必要なモジュール
PDF 保存/出力	rsvgConvert
Twitter 連携	cRUL ライブラリ [PHP]

各モジュールの導入方法については、それぞれのモジュールの解説をご確認ください

1-3 アーキテクチャ(ロジック)

1-4 アーキテクチャ(システム)

画面表示



BOK 編集作業は、各ユーザ・端末ごとに個別に作業データを持ち、編集完了時にデータを共有・反映します。

このとき、ユーザ間の編集に競合が発生した場合には、本システム独自のモジュールが自動解消します。

画面表示



このため、各 **Node** の記事編集は随時共有してます。

編集競合解消について

本システムでは、共同編集時に発生する編集競合の解消に、独自開発したモジュール (EditConflictResolver) を使用しています。

このモジュールには、以下の設定内容に基づいて、自動的に競合の解消を行います。

項目名	説明	デフォルト
BOKMERGE_PRIORITY	競合解消時に先行編集と後行編集のどちらを優先するか設定する	先行編集
BOKMERGE_DELETE_NODE_CHILD	削除する場合、削除対象の子ノードを残す/残さないを設定	残す
BOKMERGE_ADD_DELETE_CHILD	優先編集で削除されたノードに対して、非優先編集が追加したノードを残す/残さないを設定	残す
BOKMERGE_EDIT_AREA	優先される編集のロック範囲。 編集競合が発生した場合、選択範囲内の Node は削除できない。 [なし>編集対象>最小境界木>ツリー全体]	編集対象

表 1-1 編集競合解消パラメータ

以下に、具体例を示し、競合解消を説明します。

例示は、インストール直後の設定を使用した場合のものです。

設定を変更した場合の動作については、別途確認してください。

個別 Node の追加

編集前

先行編集

後行編集

[C の追加]

編集後(競合解消済み)

[D の追加]

先行編集□後行編集の順で処理し、互いの編集に影響がないため、**[両方が追加される]**

個別 Node の削除

編集前

先行編集

後行編集

[B の削除]

編集後(競合解消済み)

[A の削除]

先行編集□後行編集の順に処理し、互いの編集に影響がないため、**[両方が削除される]**

追加 Node の削除不可

編集前

先行編集

後行編集

[D の追加]

編集後(競合解消済み)

[B の削除]、[C の移動]

先行編集□後行編集の順で処理をしますが、**[D の追加]**を行う対象が**[B]**であるため、2つの後行編集の内、

[B の削除]は棄却され、**[C の移動]**のみ採用されます。

削除 Node の追加許可

編集前

先行編集

後行編集

[B の削除]

編集後(競合解消済み)

[C の追加]

先行編集□後行編集の順で処理をしますが、後行編集前に追加対象先となる**[B]**が削除されているため、

[B]の上位 Node である**[R]**へ**[C の追加]**を行う意図と読み替えます。

□2つの編集意図：**[B の削除]**、**[C の追加]**は実施済み

システム導入資料

事前確認

本書は、導入予定のサーバで、Mediawiki 及び SemanticMediaWiki が稼働していることを前提として記述しています。

それぞれの導入方法については、下記のリンクをご確認ください。

Mediawiki

Manual:Installation guide/ja - MediaWiki

SemanticMediaWiki

Help:Installation - semantic-mediawiki.org

WikiBOK

ダウンロード

本システムのソースコードは下記に、Git リポジトリもしくは圧縮ファイルとして公開されています。
下記の URL より、導入する Mediawiki の拡張フォルダ「extensions」以下にソースコードを配置してください。

<https://github.com/WikiBok/WikiBOK>

インストール

Mediawiki に拡張機能を設定します。LocalSetting.php に次の 1 行を追加してください。

```
require_once( “{$IP}/extensions/WikiBok/WikiBok.php “);
```

また、本システムでのみ利用するテーブルを作成するため、次のコマンドを実行してください。

```
$> cd maintenance
```

```
$> php -f update.php
```

PageEditToSNS

ダウンロード

本システムのソースコードは下記に、Git リポジトリもしくは圧縮ファイルとして公開されています。
下記の URL より、導入する Mediawiki の拡張フォルダ「extensions」以下にソースコードを配置してください。

<https://github.com/WikiBok/PageEditToSNS>

インストール

Mediawiki に拡張機能を設定します。LocalSetting.php に次の 1 行を追加してください。

```
require_once( “{$IP}/extensions/PageEditToSNS/PageEditToSNS.php “);
```

環境設定

本システムから Twitter(<https://twitter.com/>) への投稿を許可する必要があります。

以降の処理では、投稿用の Twitter アカунトは作成済みとします。

投稿先 Twitter ユーザの設定変更



投稿用の Twitter アカウントでログインし、「設定」→「アプリ連携」へ進みます。

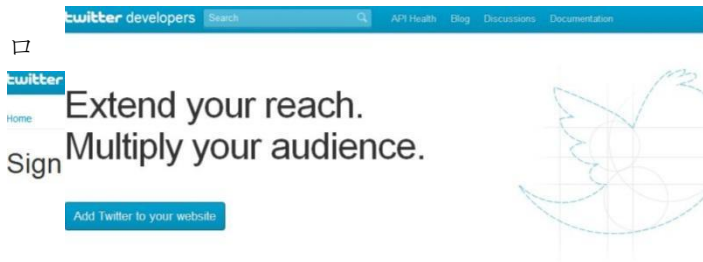
なにも設定していない場合には、右図のように「連携アプリはありません」と表示されます。

この場合には、次の「設置アプリケーションの登録」を行ってください。

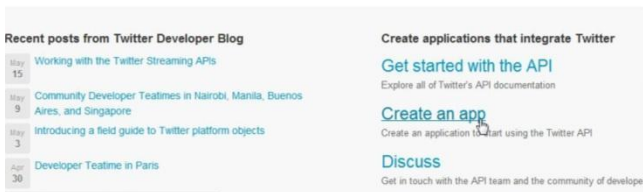
設置したアプリケーションが「連携アプリ」として設定されます。

設置アプリケーションの登録

開発者サイト(<https://dev.twitter.com/>)へアクセスし「Create an app」を選択します。



ログインしていない場合、下記、画面となるので、



投稿用の Twitter アカウントでログインします。

アプリケーション基本情報の入力

表示に従って、アプリケーションの情報を入力します。

AccessToken の取得

基本情報が表示されている画面の一番下にあるボタンをクリックします。

Your access token

It looks like you haven't authorized this application for your own Twitter account yet. For your convenience, we give you the access token here, so you can start signing your requests right away. The access token generated will reflect your application's configuration.



すると、次のように「AccessToken」が作成されます。

Your access token

Use the access token string as your "oauth_token" and the access token secret as your "oauth_token_secret" to sign your requests. Do not share your oauth_token_secret with anyone.



しかし、このままでは、Tweet の読み込みのみ許可されている状態なので、書き込みできません。

書き込みを許可

設定の変更をするためには、「Settings」タブに移動します。



その画面、中ほどに「Application Type」という項目があります。

Application Type

Access:

☐ Read only

☒ Read and Write

☐ Read, Write and Access direct messages

What type of access does your application need? Note: @Anywhere applications require read & write access.
Find out more about our [Application Permission Model](#).

これを「Read and Write」 (「Read,Write and Access direct message」でも可)に設定します。

Access Token の再取得

まだ、読み込みのみの状態なので、「Access Token」を再作成します。

Your access token

Use the access token string as your "oauth_token" and the access token secret as your "oauth_token_secret" to sign requests. Do not share your oauth_token_secret with anyone.

Access token: [REDACTED]

Access token secret: [REDACTED]

Access level: Read-only

[Recreate my access token](#)

Your access token

Use the access token string as your "oauth_token" and the access token secret as your "oauth_token_secret" to sign requests. Do not share your oauth_token_secret with anyone.

Access token: [REDACTED]

Access token secret: [REDACTED]

Access level: Read and write

[Recreate my access token](#)

再作成が完了すると「Read only」から「Read and Write」
拡張機能に登録情報を設定

サーバ側の設定ファイルの内容を変更します。

表 2-1 設定ファイル

設定内容	ファイル名[PATH]
共通設定	config/setting.php
Twitter 用設定項目	config/twitter.php
facebook 用設定項目[未実装]	config/facebook.php

表 2-2 共通設定

設定名称	設定項目
SNS_PROXY	設置サーバから SNS の API へ接続するときに必要な PROXY サーバ
SEND_TWITTER	Twitter への投稿を行う/行わない
SEND_FACEBOOK	facebook への投稿を行う/行わない

表 2-3 Twitter 設定

設定名称	設定項目
TWITTER_CONSUMER_KEY	前述□で設定された[Consumer key]
TWITTER_CONSUMER_SECRET	前述□で設定された[Consumer secret]
TWITTER_ACCESS_TOKEN	前述□で設定された[Access token]
TWITTER_ACCESS_SECRET	前述□で設定された[Access token secret]

TWITTER_USER	Twitter 投稿先のユーザ ID(上記で利用したユーザ)
TWITTER_HASH	投稿メッセージに設定する共通のハッシュタグ
TWITTER_LINK	投稿メッセージに設定するリンク(編集対象記事名称は自動補完)

上記に記載のない項目については、原則変更の必要はありません。

システム運用資料

Web 画面上での操作方法は、別紙「操作マニュアル」を参照してください。

編集操作について

BOKEditor/DescriptionEditor 上でデータ編集を行う場合、ログイン状態であることを前提としています。ユーザの作成に制限はありませんので、各自アカウントを作成してからご利用ください。

ユーザ権限の昇格

本システムで作成するユーザは、すべてノーマルユーザです。

もし特定ユーザについて、記事削除を与えたい場合には、以下の手順を用いてください。

ただし、これらの操作には、管理者としてログインする必要があることに気を付けてください。

「右上メニュー」⇒「ツールボックス」⇒「特別ページ一覧」を選択します。

「利用者と権限」にある「利用者グループの権限」を選択します。

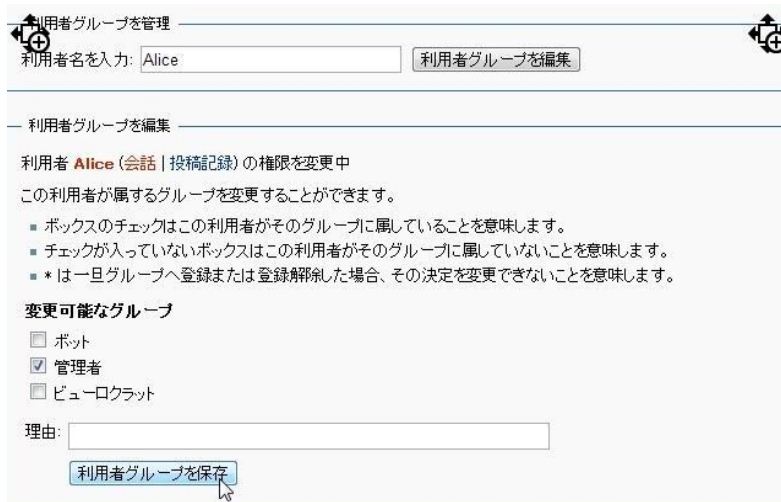


権限を変更したいユーザの ID を入力し、「利用者グループを編集」をクリック



削除権限を持ったグループを選択します。

削除権限を持ったグループはシステム管理者に確認してください(通常：管理者には削除権限があります)



処理が完了すると「所属グループ」の項目が表示されます。

利用者 **Alice** (会話 | 投稿記録) の権限を変更中

この利用者が属するグループを変更することができます。

- ボックスのチェックはこの利用者がそのグループに属していることを意味します。
- チェックが入っていないボックスはこの利用者がそのグループに属していないことを意味します。
- * は一旦グループへ登録または登録解除した場合、その決定を変更できないことを意味します。

所属グループ: **管理者**

空記事の表示について

本システムの **DescriptionEditor** 上では、一度作成された記事は、その内容が空白であったとしても、画面上に描画されます。

DescriptionEditor 上への描画そのものをやめたい場合、対象の記事を削除する必要があります。

記事の削除について

記事を削除する手順は以下の通りです。(ログインユーザに削除権限が必要です)

「右上メニュー」⇒「ツールボックス」⇒「特別ページ一覧」を選択します。

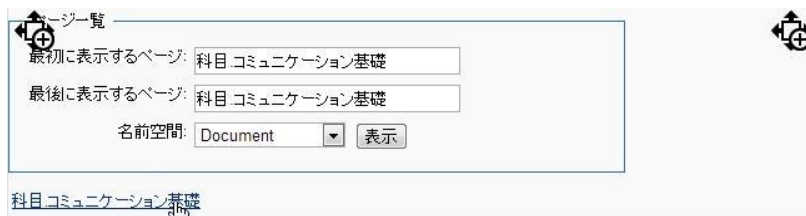
「ページの各種一覧」にある「ページ一覧」を選択します。



2つのページに削除したい記事の名前を入力し、「名前空間」から「Document」を選択し表示をクリック。



すると、「対象記事へのリンク」があるのでこれをクリック



個別の対象記事ページに移動します。

これで「右上メニュー」⇒「表示」⇒「削除」から削除を実施できます。(要：ログインユーザの削除権限)



削除理由を記述して、「ページ削除」をクリックします

警告: 削除しようとしているページには履歴があります: ページの履歴
ページをすべての履歴とともに削除しようとしています。本当にこの操作を行いたいのか、操作の結果を理解しているか、および操作が方針に従っているかどうか、確認してください。

削除理由: その他の理由 ▼

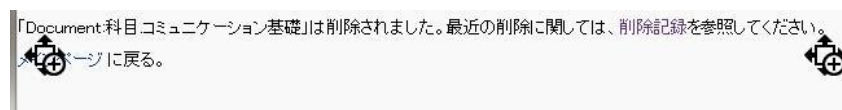
他の、または追加の理由: 内容: "[https://aguinfo.jm.aoyama.ac.jp/kouginaiyou/Shousai.aspx?f

☐ このページをウォッチする

[ページを削除](#)

該当する記録が見つかりませんでした。 [削除理由を編集する](#)

完了画面



新機能解説

ビジュアルライズモジュールの変更

画面描画用モジュールをサーバ上で動作する「Graphvis ライブラリ」から、クライアント上で動作する「d3.js ライブラリ」へ変更しました。

これに伴い、次の変化があります。

Node の折畳み/展開操作 [BOKEditor]

Description の位置変更 [DescriptionEditor]

Description 記事編集後に、リンク情報の描画を即時反映 [DescriptionEditor]

Node/Description について記事内容が空の場合の表示変更 [BOKEditor/DescriptionEditor]

Twitter への編集情報投稿

各記事の編集があった場合に、Twitter へ投稿を行う機能を実装しました。

投稿しているユーザを Follow することで、記事の更新情報をその都度取得できます。